Practitioner's Docket No. NVIDP064/P000286 *PATENT*

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of: Christopher D.S. Donham et al.

Application No.: 10/006,551 Group No.: 2674
Filed: 11/30/2001 Examiner: Nguyen, Kevin M.
For: SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR USING TEXTURES AS INSTRUCTIONS FOR GRAPHICS PROCESSING

**Mail Stop Appeal Briefs – Patents**
**Commissioner for Patents**
**P.O. Box 1450**
**Alexandria, VA 22313-1450**

## TRANSMITTAL OF APPEAL BRIEF
## (PATENT APPLICATION--37 C.F.R. § 1.192)

1. Transmitted herewith, in triplicate, is the APPEAL BRIEF in this application, with respect to the Notice of Appeal filed on March 9, 2004.

2. STATUS OF APPLICANT

This application is on behalf of other than a small entity.

---

## CERTIFICATION UNDER 37 C.F.R. §§ 1.8(a) and 1.10*

*(When using Express Mail, the Express Mail label number is **mandatory**; Express Mail certification is optional.)*

I hereby certify that, on the date shown below, this correspondence is being:

**MAILING**

X deposited with the United States Postal Service in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

**37 C.F.R. § 1.8(a)**
_ with sufficient postage as first class mail.

**37 C.F.R. § 1.10***
_ as "Express Mail Post Office to Addressee"
Mailing Label No. _____ **(mandatory)**

**TRANSMISSION**

_ facsimile transmitted to the Patent and Trademark Office, (703) _____-_____.

Melissa D. Orvis
**Signature**

Date: 4/30/04

Melissa D. Orvis
*(type or print name of person certifying)*

** Only the date of filing (' 1.6) will be the date used in a patent term adjustment calculation, although the date on any certificate of mailing or transmission under ' 1.8 continues to be taken into account in determining timeliness. See ' 1.703(f). Consider "Express Mail Post Office to Addressee" (' 1.10) or facsimile transmission (' 1.6(d)) for the reply to be accorded the earliest possible filing date for patent term adjustment calculations.*

3. FEE FOR FILING APPEAL BRIEF

Pursuant to 37 C.F.R. § 1.17(c), the fee for filing the Appeal Brief is:

other than a small entity $330.00

**Appeal Brief fee due $330.00**

4. EXTENSION OF TERM

The proceedings herein are for a patent application and the provisions of 37 C.F.R. § 1.136 apply.

Applicant believes that no extension of term is required. However, this conditional petition is being made to provide for the possibility that applicant has inadvertently overlooked the need for a petition and fee for extension of time.

5. TOTAL FEE DUE



The total fee due is:

| | |
|---|---|
| Appeal brief fee | $330.00 |
| Extension fee (if any) | $0.00 |

**TOTAL FEE DUE $330.00**

6. FEE PAYMENT

Attached is a check in the amount of $330.00.

A duplicate of this transmittal is attached.

7. FEE DEFICIENCY

If any additional extension and/or fee is required, and if any additional fee for claims is required, charge Deposit Account No. 50-1351 (Order No. NVIDP064).

Signature of Practitioner

Reg. No.: 41,429
Tel. No.: 408-971-2573
Customer No.: 28875

Kevin J. Zilka
Silicon Valley IP Group, PC
P.O. Box 721120
San Jose, CA 95172-1120
USA


**PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | |
|---|---|
| In re the application of<br><br>C. Donham et al.<br><br>Application No. 10/006,551<br><br>Filed: November 30, 2001<br><br>For: SYSTEM, METHOD AND COMPUTER PROGRAM PRODUCT FOR USING TEXTURES AS INSTRUCTIONS FOR GRAPHICS PROCESSING | Examiner: Nguyen, Kevin M.<br><br>Art Unit: 2674<br><br>Docket: NVIDP064/P000286<br><br>Date: April 30, 2004 |

**Commissioner for Patents**
**Alexandria, VA 22313-1450**

**ATTENTION: Board of Patent Appeals and Interferences**

**APPELLANT'S BRIEF (37 C.F.R. § 1.192)**

This brief is in furtherance of the Notice of Appeal, filed in this case on March 9, 2004.

The fees required under § 1.17, and any required petition for extension of time for filing this brief and fees therefor, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. § 1.192(a))

This brief contains these items under the following headings, and in the order set forth below (37 C.F.R. § 1.192(c)):

I REAL PARTY IN INTEREST

II RELATED APPEALS AND INTERFERENCES

The final page of this brief bears the practitioner's signature.

## I REAL PARTY IN INTEREST (37 C.F.R. § 1.192(c)(1))

The real party in interest in this appeal is NVIDIA Corporation.

## II RELATED APPEALS AND INTERFERENCES (37 C.F.R. § 1.192(c)(2))

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no other such appeals or interferences.

## III STATUS OF CLAIMS (37 C.F.R. § 1.192(c)(3))

### A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 1-29.

### B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims withdrawn from consideration but not canceled: None
2. Claims pending: 1-29
3. Claims allowed: None

4. Claims rejected: 1-29

## C. CLAIMS ON APPEAL

The claims on appeal are: 1-29

## IV STATUS OF AMENDMENTS (37 C.F.R. § 1.192(c)(4))

As to the status of any amendment filed subsequent to final rejection, there are no such amendments after final.

## V SUMMARY OF INVENTION (37 C.F.R. § 1.192(c)(5))

As shown in Figures 3A and 5 and the accompanying descriptions on pages 9-11 and 12-13, respectively, a system and method are provided for retrieving instructions from memory utilizing a texture module in a graphics pipeline. During use, an instruction request is sent to memory utilizing a texture module in a graphics pipeline. In response thereto, instructions are received from the memory in response to the instruction request utilizing the texture module in the graphics pipeline.

## VI ISSUES (37 C.F.R. § 1.192(c)(6))

Issue # 1: The Examiner has maintained the rejection of Claims 1-29 under 35 U.S.C. 102(e) as being anticipated by Migdal et al. (PN 6,426,753), hereinafter "Migdal."

## VII GROUPING OF CLAIMS (37 C.F.R. § 1.192(c)(7))

The claims of the above groups do not stand or fall together. Following is the grouping of claims. In the following section, appellant explains why the claims of each group are believed to be separately patentable.

Issue # 1: Grouping of Claims –

Group #1: Claims 1-6, 8, 10-17, and 20-28;

Group #2: Claim 7;

Group #3: Claim 9;

Group #4: Claims 18-19;

Group #5: Claim 29.

**VIII ARGUMENTS (37 C.F.R. § 1.192(c)(8))**

Issue #1:

The Examiner has maintained the rejection of Claims 1-29 under 35 U.S.C. 102(e) as being anticipated by Migdal et al. (PN 6,426,753), hereinafter "Migdal." Appellant respectfully disagrees with this assertion.

*Group #1: Claims 1-6, 8, 10-17, and 20-28*

With respect to the first group, the Examiner relies on the following excerpt from Migdal to show appellant's claimed "sending an instruction request to memory utilizing a texture module in a graphics pipeline" and "receiving instructions from the memory in response to the instruction request utilizing the texture module in the graphics pipeline." Note paragraph 23 on page 6 of the Examiner's Final Office Action mailed 11/26/03.

> "10. A graphics subsystem for a computer system having a distributed text memory architecture, said graphics subsystem comprising:
>
> a) a texture request generator that generates texture requests and maps said texture requests to a plurality of cache addresses, wherein said texture requests are sent to distributed texture memories of said computer system according to a first ordering;
>
> b) an address queue for receiving and storing said plurality of cache addresses according to said first

> ordering;
>
> c) a cache memory for receiving texture responses from said distributed texture memories, wherein said texture responses enter said cache memory according to a second ordering; and
>
> d) a texture filter for performing texture filtering by retrieving said texture responses from said cache memory in an order corresponding to said first ordering and independent of said second ordering." (col. 14, lines 50-65)

The Examiner continues by stating that Migdal teaches a texture request being sent to distributed texture memories of a computer, and cache memory for receiving texture responses from distributed texture memories. Further, the Examiner asserts that, in a computer system, there are always instructions to input, output and process data (instruction request).

With respect to the Examiner's statement regarding what is "always ... in a computer system," it appears that the Examiner has simply invoked Official Notice regarding appellant's claimed "instruction" limitation in the context of the remaining limitations. In response, applicant had explained the paramount benefits coinciding with the foregoing claim language (which are absent in Migdal), as well as formally requested a specific showing of the subject matter in ALL of the claims. Note excerpt from MPEP below.

> "If the applicant traverses such an [Official Notice] assertion the examiner should cite a reference in support of his or her position." See MPEP 2144.03.

The Examiner has failed to meet this request to date.

A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. Of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). Moreover, the identical invention must be shown in as complete detail as contained in the claim. *Richardson v. Suzuki Motor Co.*868 F.2d

1226, 1236, 9USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990).

This criterion has simply not been met by the Examiner's rejection. Specifically, only appellant teaches and claims sending "instructions requests" and retrieving "instructions" in the specific context of "utilizing a texture module."

Even if the Examiner made such a showing, however, appellant contends that it would not be obvious to modify Migdal to include appellant's claimed management of an "instruction request" and "instruction" "utilizing a texture module." By retrieving the instructions utilizing the texture module, much pipeline bandwidth is saved at the input of the texture module, since prior art configuration data at least in part need not necessarily be received from the rasterizer. Moreover, the memory traditionally employs a high-bandwidth connection with the texture module, which may be used for efficient retrieval of the instructions.

The instructions may then be used by the texture module in order to control various graphics processing involving the texels, pixels, and/or primitives, etc. For example, the instructions may control how subsequent texels may be mapped to pixels associated with primitives. Moreover, the instructions may be used to control the mapping, or blending, of the texels with the pixels, in accordance with the instructions. Simply nowhere in the prior art is there such a combination of features for fulfilling the foregoing objectives.

*Group #2: Claim 7*

Regarding Claim 7, appellant claims "wherein the instructions control the manner in which the texture environment module processes the texture information." The Examiner relies on col. 5, lines 15-20 of Migdal to show such feature in the prior art. See below.

> "The G chip 105 accepts instructions and data from Crosstalk streams 106. The instructions are executed by microprocessor 103 and G chip 105. G chip 105 also performs geometric calculations on vertex data. Data is temporarily cached in SRAM 104. Eventually, the resulting vertex data is sent over the high bandwidth network 102 to one of the R subsystems." (col. 5, lines 15-20)

After a careful review, appellant contends that such excerpt is lacking. For example, the abovementioned "microprocessor" and "G chip" do not include a "texture environment module," as claimed by appellant. Only appellant teaches and claims sending "instructions requests" and retrieving "instructions" "utilizing a texture module" to control the manner in which the texture environment module processes the texture information, which provides the aforementioned advantages that are non-existent in the prior art including Migdal.

Thus, the aforementioned anticipation criterion has simply not been met by the Examiner's proposed excerpt.

*Group #3: Claim 9*

With respect to Claim 9, the Examiner has attempted to make a prior art showing of appellant's claimed "initial instructions [that] control at least the sending of the instruction request by the texture module" using the following excerpts from Migdal.

> "As stated above, the R chip 501 accepts geometric and pixel primitives from the network and renders each one into a specified 16.times.16 screen patch. The R chip 501 also handles requests from other R chips for texture data and requests from the display subsystem for frame buffer data. Attached to this are sixteen M (multisample) memory chips, which are used to store the frame buffer screen patches corresponding to the subsystem." (col. 7, lines 55-60)

Again, the aforementioned anticipation criterion has simply not been met by the Examiner's reference. For example, there is simply no disclosure, teaching, or even

suggestion of "initial instructions" that specifically control the sending of an "instruction request by the texture module."

*Group #4: Claims 18-19*

With respect to Claims 18-19, the Examiner has attempted to make a prior art showing of appellant's claimed "complete instruction set [that] is received in response to the instruction request" (see Claim 18), and "partial instruction set [that] is received in response to the instruction request" (see Claim 19). The Examiner relies on the following excerpts from Migdal to show such features in the prior art.

> "Texture request generator 603 also maps the texture coordinates to a cache address and sends the cache address to FIFO memory 604, which acts as a queue for cache addresses. Cache addresses received by FIFO memory 604 will reach texture filter 605 after multiple clock cycles. Texture filter 605, upon receiving the cache addresses, will retrieve the texture responses from the texture response buffer 608." (col. 9, lines 30-37)

Still yet, the aforementioned anticipation criterion has simply not been met by the Examiner's reference. There is simply no disclosure, teaching, or even suggestion of a complete or partial instruction set being received in response to the "instruction request," as claimed in combination with the remaining claim elements.

*Group #5: Claim 29*

With respect to Claim 29, the Examiner has attempted to make a prior art showing of the following limitations using the shown Migdal excerpts.

Such mapping, however, fails in many respects. Specifically, the aforementioned anticipation criteria has simply not been met by the Examiner's reference, as indicated by <u>at least</u> the **bolded** limitations shown below.

| **Excerpts from Claim 29** | **Excerpts from Migdal relied upon by the Examiner** |
|---|---|

| | |
|---|---|
| receiving a plurality of preliminary instructions from a rasterizer module utilizing a shader module coupled thereto; | "The G chip 105 accepts instructions and data from Crosstalk streams 106. The instructions are executed by microprocessor 103 and G chip 105. G chip 105 also performs geometric calculations on vertex data. Data is temporarily cached in SRAM 104. Eventually, the resulting vertex data is sent over the high bandwidth network 102 to one of the R subsystems.<br><br>In the currently preferred embodiment, two to sixteen R subsystems 107 are used to perform pixel rasterization functions. R subsystems basically are comprised of a rasterization ASIC (R) chip 108 and multiple memory (M) chips 109. The actual rasterization functions are performed by R chip 108, whereas framebuffer and texture memory is provided by the M chips." (col. 5, lines 15-30) |
| **sending an instruction request to memory utilizing a texture module coupled to the shader module;** | "As stated above, the R chip 501 accepts geometric and pixel primitives from the network and renders each one into a specified 16.times.16 screen patch. The R chip 501 also handles requests from other R chips for texture data and requests from the display subsystem for frame buffer data. Attached to this are sixteen M (multisample) memory chips, which are used to store the frame buffer screen patches corresponding to the subsystem." (col. 7, lines 55-60) |
| **receiving additional instructions from the memory in response to the instruction request utilizing the texture module;** | "According to the present embodiment, a VALID bit is asserted when texture data stored in an associated cache line (e.g., cache line 1210b) is ready to be retrieved by texture filter 605." (col. 9, lines 64-67) |
| **caching the additional instructions on the texture module;** | "FIG. 12 is a diagram of a cache memory 1200 within texture response buffer 608 according to an embodiment of the present invention. As illustrated, cache memory 1200 includes a plurality of cache lines 1210a-1210h for storing texture data." (col. 9, lines 55-60) |
| sending a texture request to memory utilizing the texture module **in accordance with the additional instructions;** | "The texture request generator 603 receives one, two, or three texture coordinates for a pixel (corresponding to 1D, 2D, or 3D texture) and generates texture requests for texture data that needs to be loaded into the texture response buffer 608 in order to process the pixel. Texture request generator 603 also maps the texture coordinates to a cache address and sends the cache address to FIFO memory 604, which acts as a queue for cache addresses. Cache addresses received by FIFO memory 604 will reach texture filter 605 after multiple clock cycles. Texture filter 605, upon receiving the cache addresses, will retrieve the texture responses from the texture response buffer 608." (col. 9, lines 25-35) |
| receiving texture information from the memory in response to the texture request utilizing the texture module; | "Texture request generator 603 also maps the texture coordinates to a cache address and sends the cache address to FIFO memory 604, which acts as a queue for cache addresses. Cache addresses received by FIFO memory 604 will reach texture filter 605 after multiple clock cycles. Texture filter 605, upon receiving the cache addresses, will retrieve the texture responses from the texture response buffer 608." (col. 9, lines 30-35) |
| caching the texture information on the texture module; | "FIG. 12 is a diagram of a cache memory 1200 within texture response buffer 608 according to an embodiment of the present invention. As illustrated, cache memory 1200 includes a plurality of cache lines 1210a-1210h for storing texture data." (col. 9, lines 55-60) |
| processing a plurality of pixels with the texture information utilizing the shader module **in accordance with the additional instructions;** | "The R chip is comprised of the message receive unit 601; scan converter 602; texture request generator 603; texture FIFO 604; texture filter and imaging unit 605; lighting, texture, and fog unit 606; fragment unit 607; texture response buffer 608; message transmit unit 609; network interface 610; FIFO write unit 611; FIFO read unit 612; texture/pixel response processor 613; display response processor 614; and M input/output (I/O) controller 615. The message receive unit 601 accepts packets from the |

| | |
|---|---|
| | input network interface 610 and delivers them to the appropriate block within the R chip. The scan converter 602 scan converts and z-rejects primitive packets and also generates texture coordinates for the texture request generator 603. The texture request generator 603 receives one, two, or three texture coordinates for a pixel and generates the required tile addresses for tiles that need to be loaded into the texture response buffer 608 in order to process the pixel. The texture FIFO 604 stores data needed by the texture response buffer 608 and the texture filter and imaging unit 605 in order to filter texture. The data includes cache addresses, tile addresses, and filter coefficients. The texture FIFO 604 is sized to handle the latency of a texture request/response. Texture filter and imaging unit 605 performs the necessary filtering for texture mapping and image processing including convolution and texture look-up table support. A lighting, texture, and fog unit 606 accepts pixels from the scan converter 602 and filtered texture from the texture filter and imaging unit 605 and performs per-pixel lighting, texture environment, and fog computation. Fragments from the lighting, texture, and fog unit 606 are input to fragment unit 607 which redistributes the fragments to the M chips for processing. It also handles all the buffering of fragment, download, and register packets to the M chips." (col. 8, lines 25-55) |
| repeating (b) – (h) **in accordance with the additional instructions;** and<br><br>outputting the processed pixels **upon receipt of additional instructions that include a terminate instruction.** | "If it is determined, at step 806, that the value of the COUNT bits is not zero, it can be understood that the cache line may not be available for receiving new texture data. It can also be understood that the cache line already contains the required texture data. Thus, at step 808, the texture request generator 603 checks the count-and-tag memory 1100 and determines whether the TAG bits match the texture coordinates.<br><br>If it is determined that the TAG bits (within count-and-tag memory 1100) corresponding to the specified cache line match the texture coordinates, then it could be understood that the cache line already contains the required texture data. Accordingly, it is not necessary to request the same texture data again from the network. Thus, steps 812 and 814 are carried out to update the count-and-tag memory 1100. Thereafter, the texture request generator 603 is ready to repeat process 800 beginning with step 802 for another pair of texture coordinates." (col. 11, lines 18-35) |

In view of the remarks set forth hereinabove, all of the independent claims are deemed allowable, along with any claims depending therefrom.

## IX APPENDIX OF CLAIMS (37 C.F.R. § 1.192(c)(9))

The text of the claims involved in the appeal is:

1. (Original) A method for retrieving instructions from memory utilizing a texture module in a graphics pipeline, comprising:

(a) sending an instruction request to memory utilizing a texture module in a graphics pipeline; and

(b) receiving instructions from the memory in response to the instruction request utilizing the texture module in the graphics pipeline.

2. (Original) The method as recited in claim 1, and further comprising sending a texture request to memory utilizing the texture module in the graphics pipeline.

3. (Original) The method as recited in claim 2, and further comprising receiving texture information from the memory in response to the texture request utilizing the texture module in the graphics pipeline.

4. (Original) The method as recited in claim 1, wherein the memory includes a frame buffer.

5. (Original) The method as recited in claim 4, wherein the memory includes direct random access memory (DRAM).

6. (Original) The method as recited in claim 3, wherein the instructions are adapted for controlling a texture environment module coupled to the texture module.

7. (Original) The method as recited in claim 6, wherein the instructions control the manner in which the texture environment module processes the texture information.

8. (Original) The method as recited in claim 1, and further comprising receiving initial instructions from a rasterizer module coupled to the texture module.

9. (Original) The method as recited in claim 8, wherein the initial instructions control at least the sending of the instruction request by the texture module.

10. (Original) The method as recited in claim 3, and further comprising temporarily storing the instructions and the texture information in cache.

11. (Original) The method as recited in claim 10, wherein the cache is resident on the texture module.

12. (Original) The method as recited in claim 3, wherein each piece of texture information and each of the instructions are of a similar size in the memory.

13. (Original) The method as recited in claim 3, and further comprising controlling the texture module utilizing a shader module coupled thereto.

14. (Original) The method as recited in claim 13, wherein the shader module controls the sending of the instruction request and the texture request by the texture module.

15. (Original) The method as recited in claim 13, wherein the shader module processes a plurality of pixels with the texture information based on the instructions.

16. (Original) The method as recited in claim 15, wherein the shader module is capable of reusing the texture information in order to request further texture information from the memory.

17. (Original) The method as recited in claim 15, and further comprising ceasing the processing upon the receipt of a terminate instruction.

18. (Original) The method as recited in claim 1, wherein a complete instruction set is received in response to the instruction request.

19. (Original) The method as recited in claim 1, wherein a partial instruction set is received in response to the instruction request.

20. (Original) The method as recited in claim 19, and further comprising repeating (a) – (b) in accordance with the instructions.

21. (Original) The method as recited in claim 1, wherein (a) – (b) are carried out in accordance with the instructions received in response to the instruction request.

22. (Original) The method as recited in claim 1, wherein the texture module is adapted for operating in a plurality of different modes.

23. (Original) The method as recited in claim 22, wherein the instructions are received in a predetermined one or more of the different modes.

24. (Original) A computer program product for retrieving instructions from memory utilizing a texture module in a graphics pipeline, comprising:

(a) computer code for sending an instruction request to memory utilizing a texture module in a graphics pipeline; and

(b) computer code for receiving instructions from the memory in response to the instruction request utilizing the texture module in the graphics pipeline.

25. (Original) A system for retrieving instructions from memory utilizing a texture module in a graphics pipeline, comprising:

(a) means for sending an instruction request to memory; and

(b) means for receiving instructions from the memory in response to the instruction request.

26. (Original) A texture module for retrieving instructions from memory capable of carrying out a method, comprising:

(a) sending an instruction request to memory; and

(b) receiving instructions from the memory in response to the instruction request.

27. (Original) A data structure stored in a frame buffer of a graphics pipeline for allowing the retrieval of instructions utilizing a texture module coupled thereto, comprising an instruction object stored in the frame buffer for being retrieved therefrom in response to an instruction request utilizing a texture module in a graphics pipeline.

28. (Original) A method for retrieving instructions from memory, comprising:

(a) receiving a plurality of preliminary instructions from a rasterizer module utilizing a texture module coupled thereto;

(b) sending an instruction request to memory utilizing the texture module;

(c) receiving additional instructions from the memory in response to the instruction request utilizing the texture module;

(d) caching the additional instructions on the texture module;

(e) sending a texture request to memory utilizing the texture module in accordance with the additional instructions;

(f) receiving texture information from the memory in response to the texture request utilizing the texture module;

(g) caching the texture information on the texture module; and

(h) repeating (b) – (g) in accordance with the additional instructions.

29. (Original) A method for retrieving instructions from memory, comprising:

(a) receiving a plurality of preliminary instructions from a rasterizer module utilizing a shader module coupled thereto;

(b) sending an instruction request to memory utilizing a texture module coupled to the shader module;

(c) receiving additional instructions from the memory in response to the instruction request utilizing the texture module;

(d) caching the additional instructions on the texture module;

(e) sending a texture request to memory utilizing the texture module in accordance with the additional instructions;

(f) receiving texture information from the memory in response to the texture request utilizing the texture module;

(g) caching the texture information on the texture module;

(h) processing a plurality of pixels with the texture information utilizing the shader module in accordance with the additional instructions;

(i) repeating (b) – (h) in accordance with the additional instructions; and

(j) outputting the processed pixels upon receipt of additional instructions that include a terminate instruction.

In the event a telephone conversation would expedite the prosecution of this application, the Examiner may reach the undersigned at (408) 971-2573. For payment of any additional fees due in connection with the filing of this paper, the Commissioner is authorized to charge such fees to Deposit Account No. 50-1351 (Order No. NVIDP064_P000286).

Respectfully submitted,

By: _____ Date: 04/30/04

Kevin J. Zilka

Reg. No. 41,429

Silicon Valley IP Group, P.C.
P.O. Box 721120
San Jose, California 95172-1120
Telephone: (408) 971-2573
Facsimile: (408) 971-4660